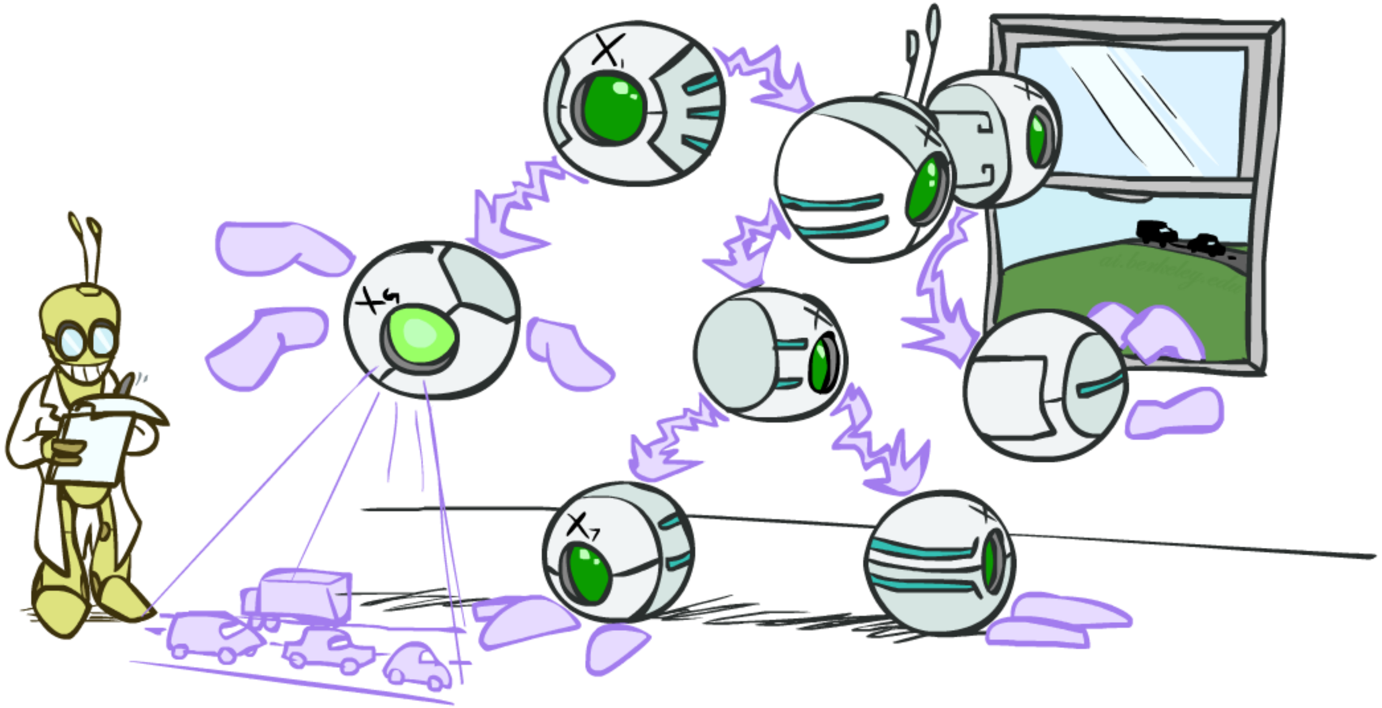
CS 188: Artificial Intelligence

Bayes Nets: Exact Inference



Instructors: Angela Liu and Yanlai Yang

University of California, Berkeley

(Slides adapted from Stuart Russell and Dawn Song)

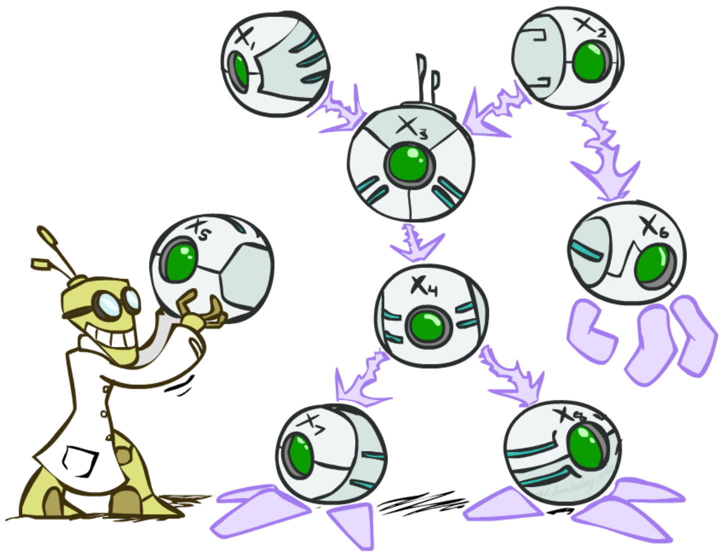
Bayes Net Semantics

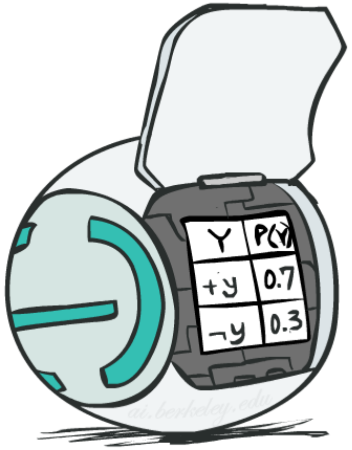
- A Bayes net is an efficient encoding of a probabilistic model of a domain
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

$$P(X|a_1 \ldots a_n)$$

- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\textit{parents}(X_i))$$





Size of a Bayes Net

- How big is a joint distribution over N Boolean variables?

  2^N

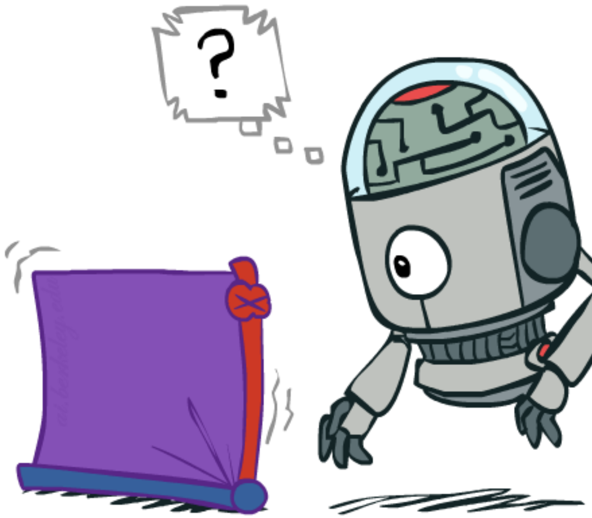
- How big is an N-node net if nodes have up to k parents?

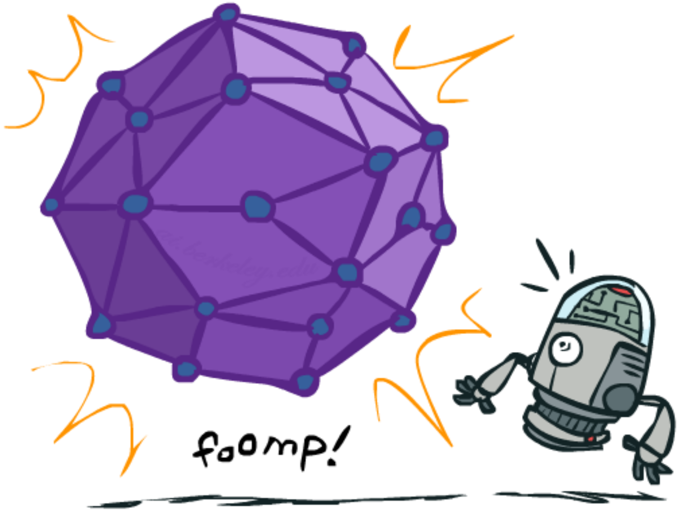
  $O(N * 2^{k+1})$

- Both give you the power to calculate

  $$P(X_1, X_2, \ldots X_n)$$

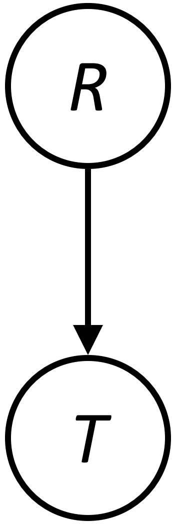
- BNs: Huge space savings!
- Also easier to elicit local CPTs
- Also faster to answer queries (this lecture!)





Causality?

- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables *Traffic* and *Rain*

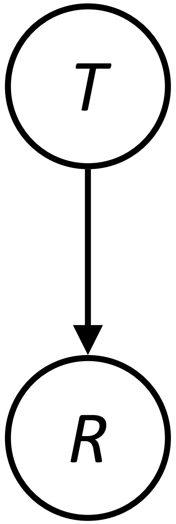


$P(R)$

+r	1/4
-r	3/4

$P(T|R)$

+r	+t	3/4
	-t	1/4
-r	+t	1/2
	-t	1/2



$P(T)$

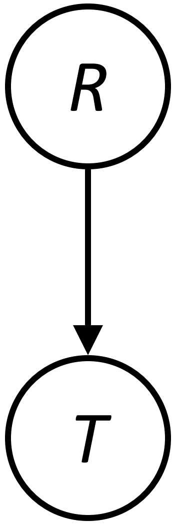
+t	9/16
-t	7/16

$P(R|T)$

+t	+r	1/3
	-r	2/3
-t	+r	1/7
	-r	6/7

Example: Traffic

- Causal direction



$P(R)$

+r	1/4
-r	3/4

$P(T|R)$

+r	+t	3/4
	-t	1/4

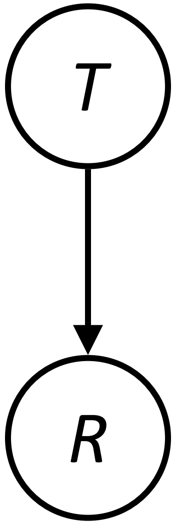
-r	+t	1/2
	-t	1/2

$P(T, R)$

+r	+t	3/16
+r	-t	1/16
-r	+t	6/16
-r	-t	6/16

Example: Reverse Traffic

- Reverse causality?



$P(T)$

+t	9/16
-t	7/16

$P(R|T)$

+t	+r	1/3
	-r	2/3
-t	+r	1/7
	-r	6/7

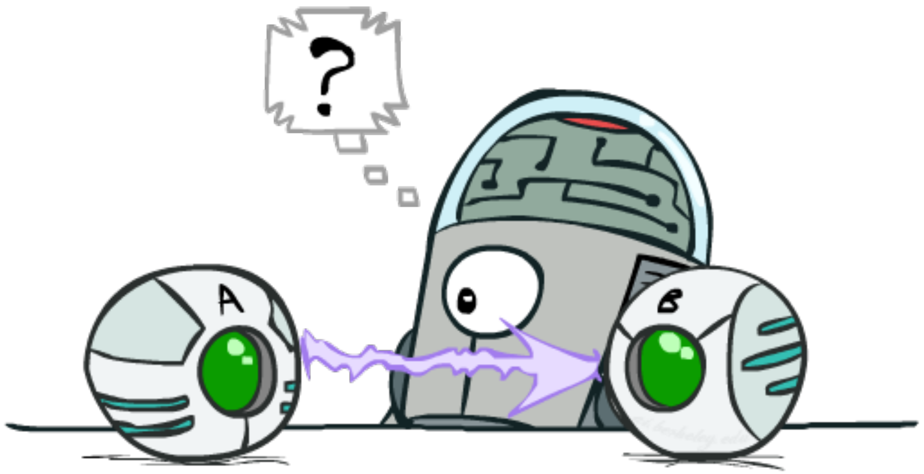
$P(T, R)$

+r	+t	3/16
+r	-t	1/16
-r	+t	6/16
-r	-t	6/16

Causality?

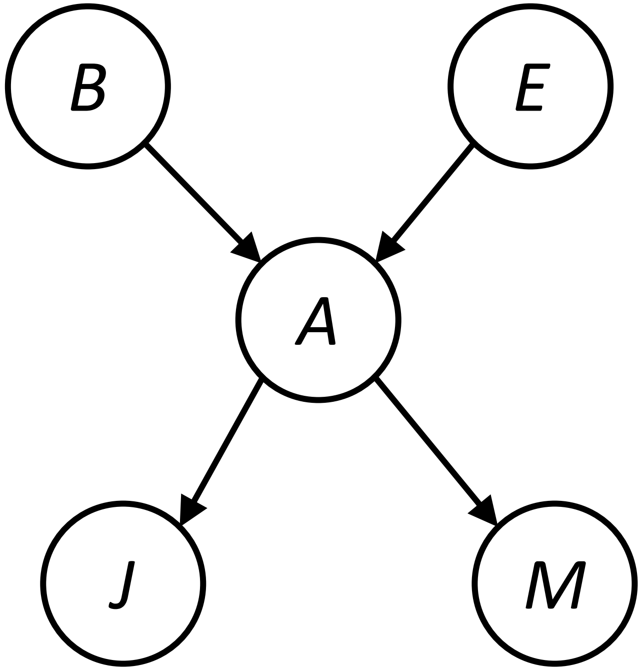
- When Bayes' nets reflect the true causal patterns:
  - Often simpler (nodes have fewer parents)
  - Often easier to think about
  - Often easier to estimate probabilities from data
- BNs need not actually be causal
  - Sometimes no causal net exists over the domain (especially if variables are missing)
  - E.g. consider the variables *Traffic* and *Drips*
  - End up with arrows that reflect correlation, not causation
- What do the arrows really mean?
  - Topology may happen to encode causal structure
  - Topology really encodes conditional independence

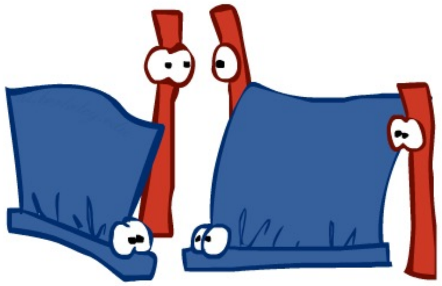
$$P(x_i|x_1, \ldots x_{i-1}) = P(x_i|\mathit{parents}(X_i))$$

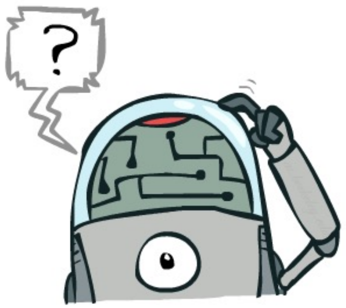


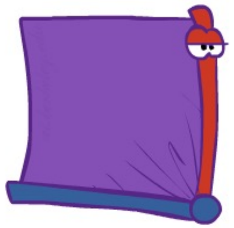
Inference by Enumeration in Bayes Net

- Reminder of inference by enumeration:
  - Any probability of interest can be computed by summing entries from the joint distribution: $P(\mathbf{Q} \mid \mathbf{e}) = \alpha \sum_{h} P(\mathbf{Q}, \mathbf{h}, \mathbf{e})$
  - Entries from the joint distribution can be obtained from a BN by multiplying the corresponding conditional probabilities
- $P(B \mid j, m) = \alpha \sum_{e,a} P(B, e, a, j, m)$
- $= \alpha \sum_{e,a} P(B)\, P(e)\, P(a \mid B,e)\, P(j \mid a)\, P(m \mid a)$
- Problem: sums of ***exponentially many*** products!



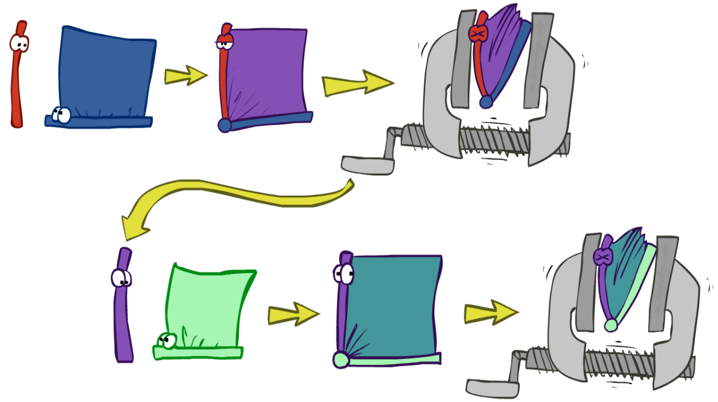






Variable elimination: The basic ideas

- Move summations inwards as far as possible
  - $P(B \mid j, m) = \alpha \sum_{e,a} P(B)\, P(e)\, P(a|B,e)\, P(j|a)\, P(m|a)$
  - $= \alpha\, P(B) \sum_{e} P(e) \sum_{a} P(a|B,e)\, P(j|a)\, P(m|a)$



- Do the calculation from the inside out
  - I.e., sum over *a* first, then sum over *e*
  - Challenge: $P(a|B,e)$ isn't a single number, it's a table of different numbers (depending on the values of *B* and *e*)
  - Solution: use arrays of numbers with appropriate operations on them; these are called ***factors***

Factor Zoo I

- Joint distribution: P(X,Y)
  - Entries P(x,y) for all x, y
  - |X|x|Y| matrix
  - Sums to 1

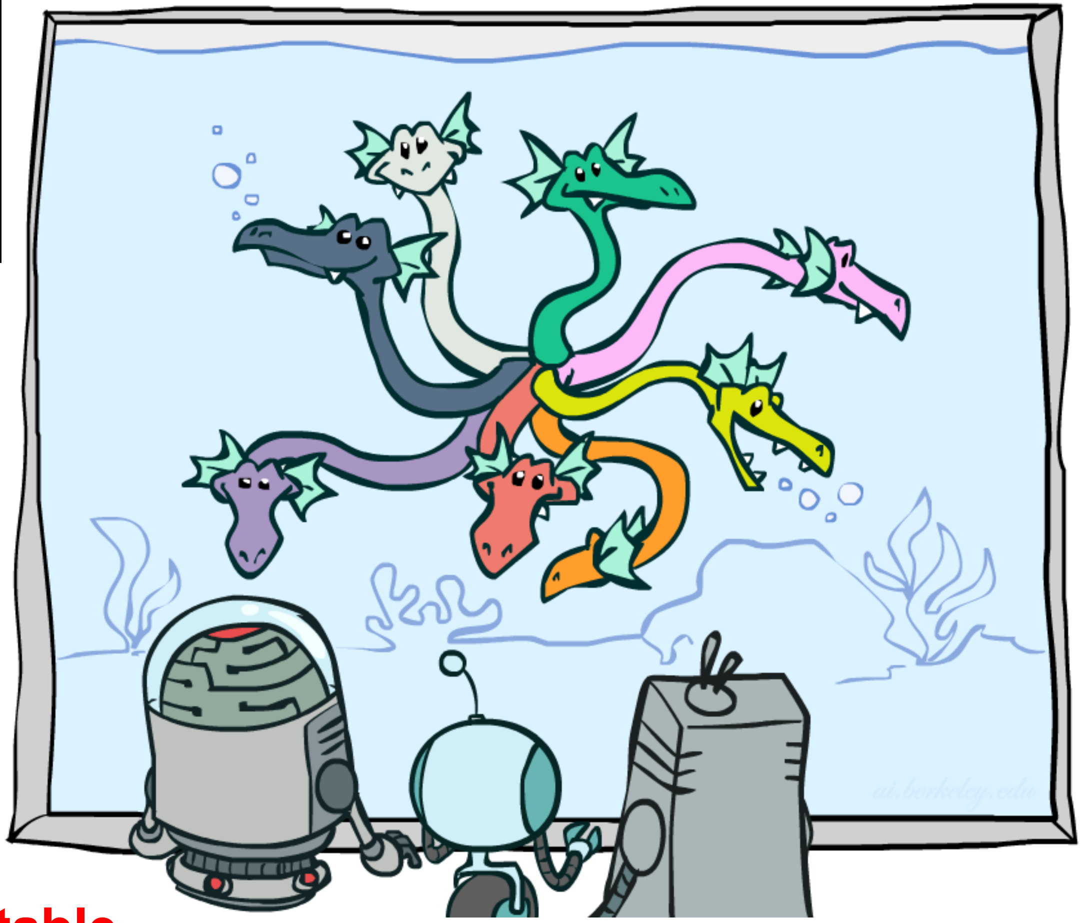
P(A,J)

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

- Projected joint: P(x,Y)
  - A slice of the joint distribution
  - Entries P(x,y) for one x, all y
  - |Y|-element vector
  - Sums to P(x)

$P(a,J) = P_a(J)$

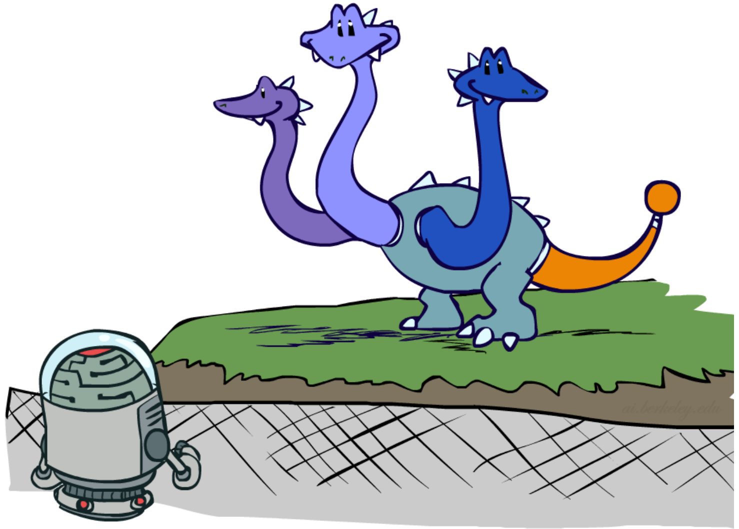
A \ J	true	false
true	0.09	0.01



Number of variables (capitals) = dimensionality of the table

Factor Zoo II

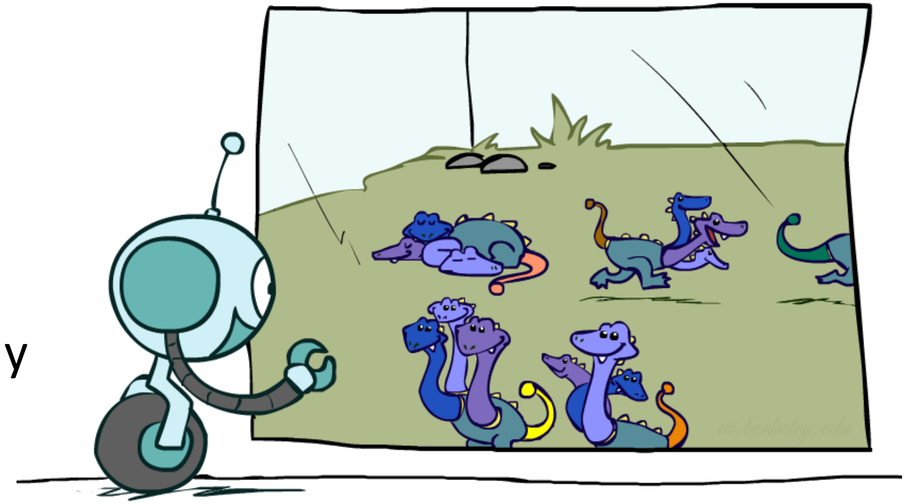
- Single conditional: P(Y | x)
  - Entries P(y | x) for fixed x, all y
  - Sums to 1



$P(J|a)$

A \ J	true	false
true	0.9	0.1

- Family of conditionals: P(X |Y)
  - Multiple conditionals
  - Entries P(x | y) for all x, y
  - Sums to |Y|

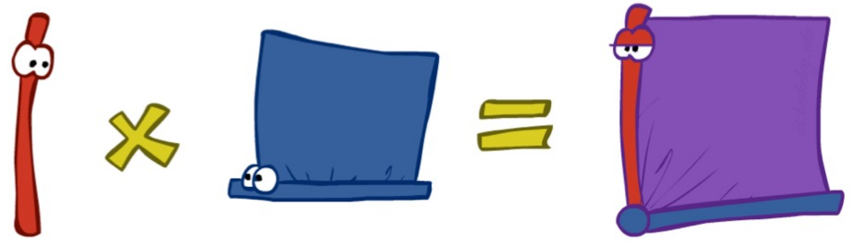


$P(J|A)$

A \ J	true	false	
true	0.9	0.1	} - $P(J|a)$
false	0.05	0.95	} - $P(J|\neg a)$

Operation 1: Pointwise product

- First basic operation: ***pointwise product*** of factors (***not*** matrix multiply!)
  - New factor has ***union*** of variables of the two original factors
  - Each entry is the product of the corresponding entries from the original factors



- Example: $P(J|A)$ x $P(A)$ = $P(A,J)$

$P(A)$

true	0.1
false	0.9

x

$P(J|A)$

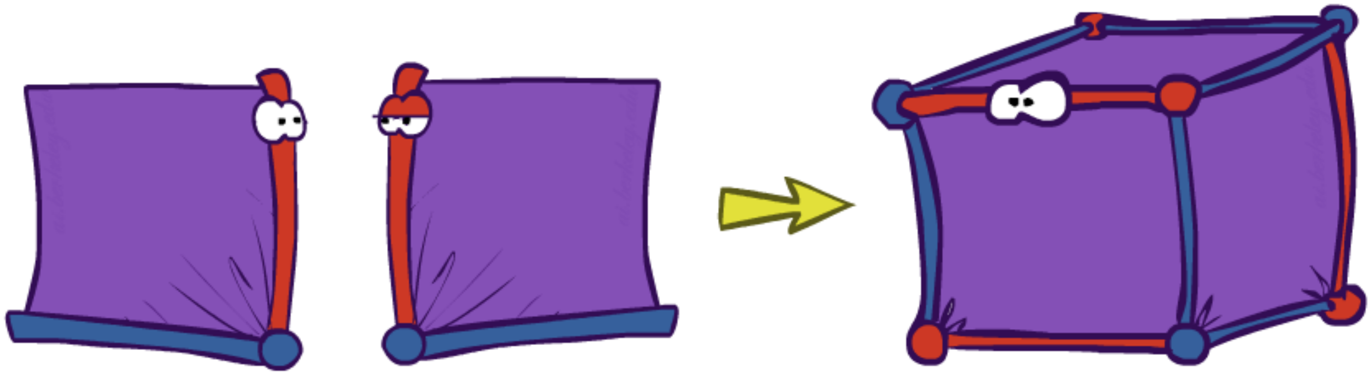
A \ J	true	false
true	0.9	0.1
false	0.05	0.95

=

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

Example: Making larger factors



- Example: $P(A,J) \times P(M|A) = P(A,J,M)$
- Factor blowup can make VE very expensive!

$P(A,J)$

A \ J	true	false
true	0.09	0.01
false	0.045	0.855

x

$P(M \mid A)$

A \ M	true	false
true	0.7	0.3
false	0.01	0.99

=

$P(A,J,M)$

A=true

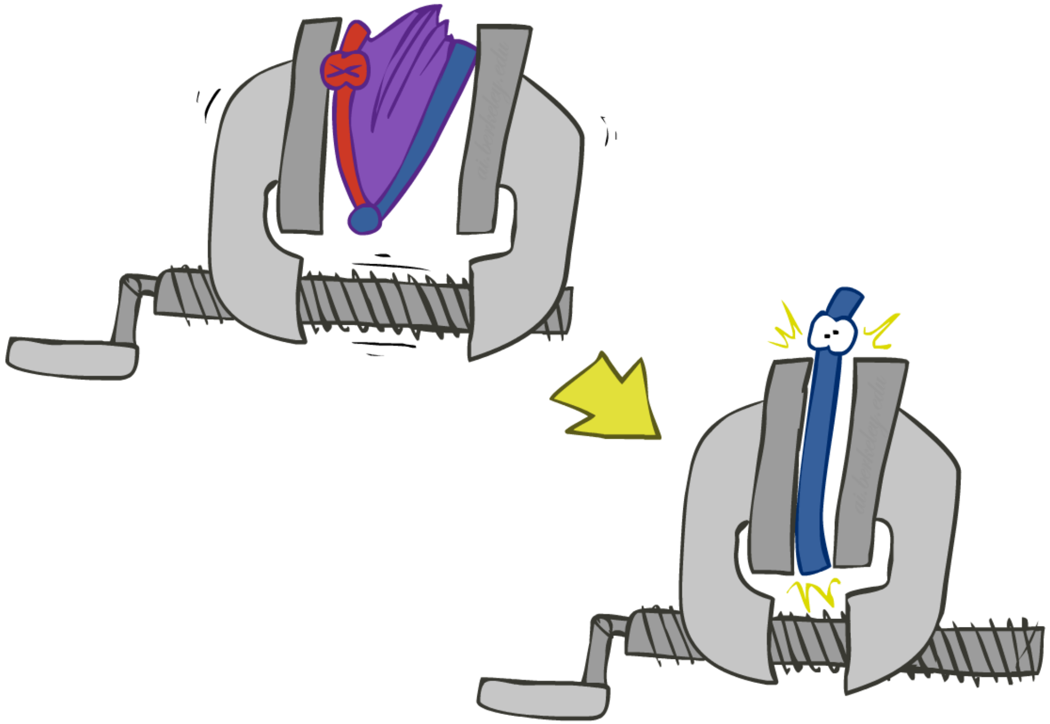
J \ M	true	false
true		
false		0.003

A=false

J \ M	true	false
		0.846

Operation 2: Summing out a variable

- Second basic operation: ***summing out*** (or eliminating) a variable from a factor
  - Shrinks a factor to a smaller one
- Example: $\sum_j P(A,J) = P(A,j) + P(A,\neg j) = P(A)$



$P(A,J)$

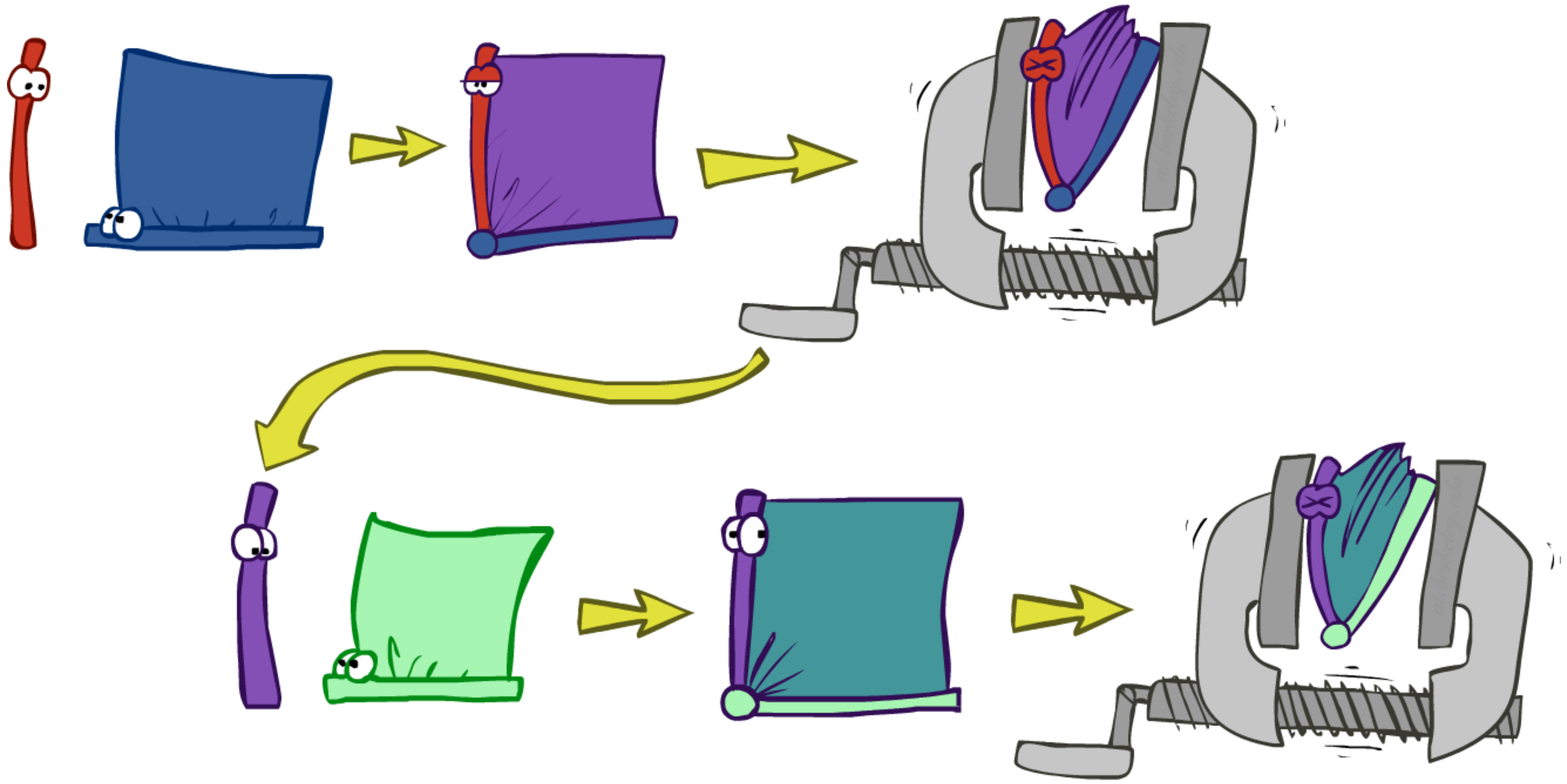
A \ J	true	false
true	0.09	0.01
false	0.045	0.855

Sum out J →

$P(A)$

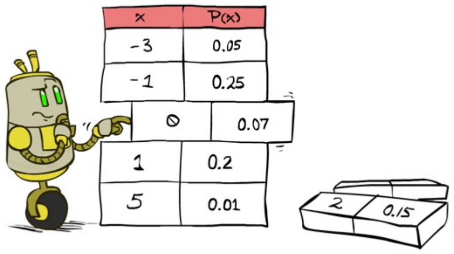
true	0.1
false	0.9

Variable Elimination

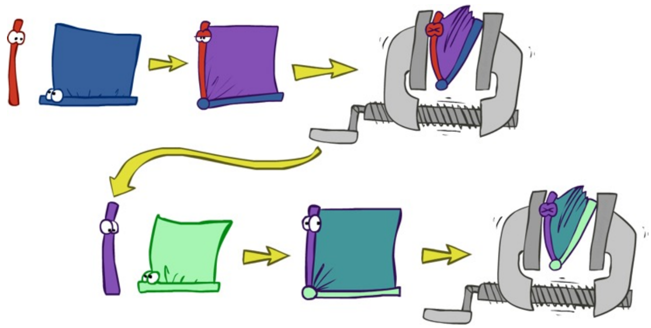


Variable Elimination

- Query: $P(Q|E_1=e_1,.., E_k=e_k)$

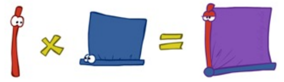


- Start with initial factors:
  - Local CPTs (but instantiated by evidence)



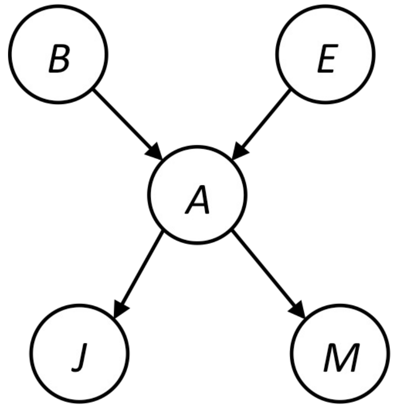
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H_j
  - Eliminate (sum out) H_j from the product of all factors mentioning H_j

- Join all remaining factors and normalize

 $\times\ \alpha$

Example

Query $P(B \mid j,m)$



$P(B)$	$P(E)$	$P(A \mid B,E)$	$P(j \mid A)$	$P(m \mid A)$

Choose A

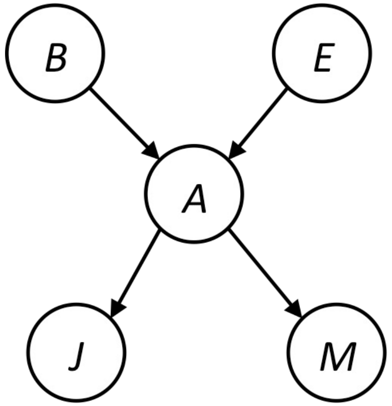
$P(A \mid B,E)$
$P(j \mid A)$
$P(m \mid A)$

$\times$ ⇒ Σ ⇒ $P(j,m \mid B,E)$

$P(B)$	$P(E)$	$P(j,m \mid B,E)$

Example

$P(B)$ $P(E)$ $P(j,m|B,E)$



Choose E

$P(E)$
$P(j,m|B,E)$ → $\times$ → Σ → $P(j,m|B)$

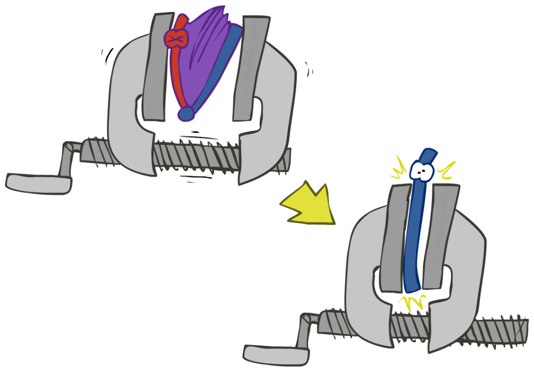
$P(B)$ $P(j,m|B)$

Finish with B

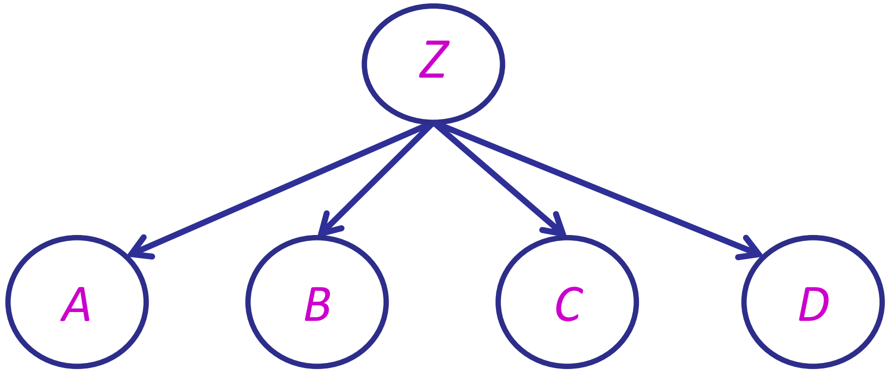
$P(B)$
$P(j,m|B)$ → $\times$ → $P(j,m,B)$ → Normalize → $P(B \mid j,m)$

Summing out from a product of factors

- Project the factors each way first, then sum the products
- Example: $\sum_a P(a|B,e) \times P(j|a) \times P(m|a)$
- $= P(a|B,e) \times P(j|a) \times P(m|a) +$
- $P(\neg a|B,e) \times P(j|\neg a) \times P(m|\neg a)$



Order matters



- Order the terms Z, A, B C, D
  - $P(D) = \alpha \sum_{z,a,b,c} P(z)\, P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)$
  - $\quad = \alpha \sum_z P(z) \sum_a P(a|z) \sum_b P(b|z) \sum_c P(c|z)\, P(D|z)$
  - Largest factor has 2 variables (D,Z)
- Order the terms A, B C, D, Z
  - $P(D) = \alpha \sum_{a,b,c,z} P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - $\quad = \alpha \sum_a \sum_b \sum_c \sum_z P(a|z)\, P(b|z)\, P(c|z)\, P(D|z)\, P(z)$
  - Largest factor has 4 variables (A,B,C,D)
  - In general, with n leaves, factor of size 2^n

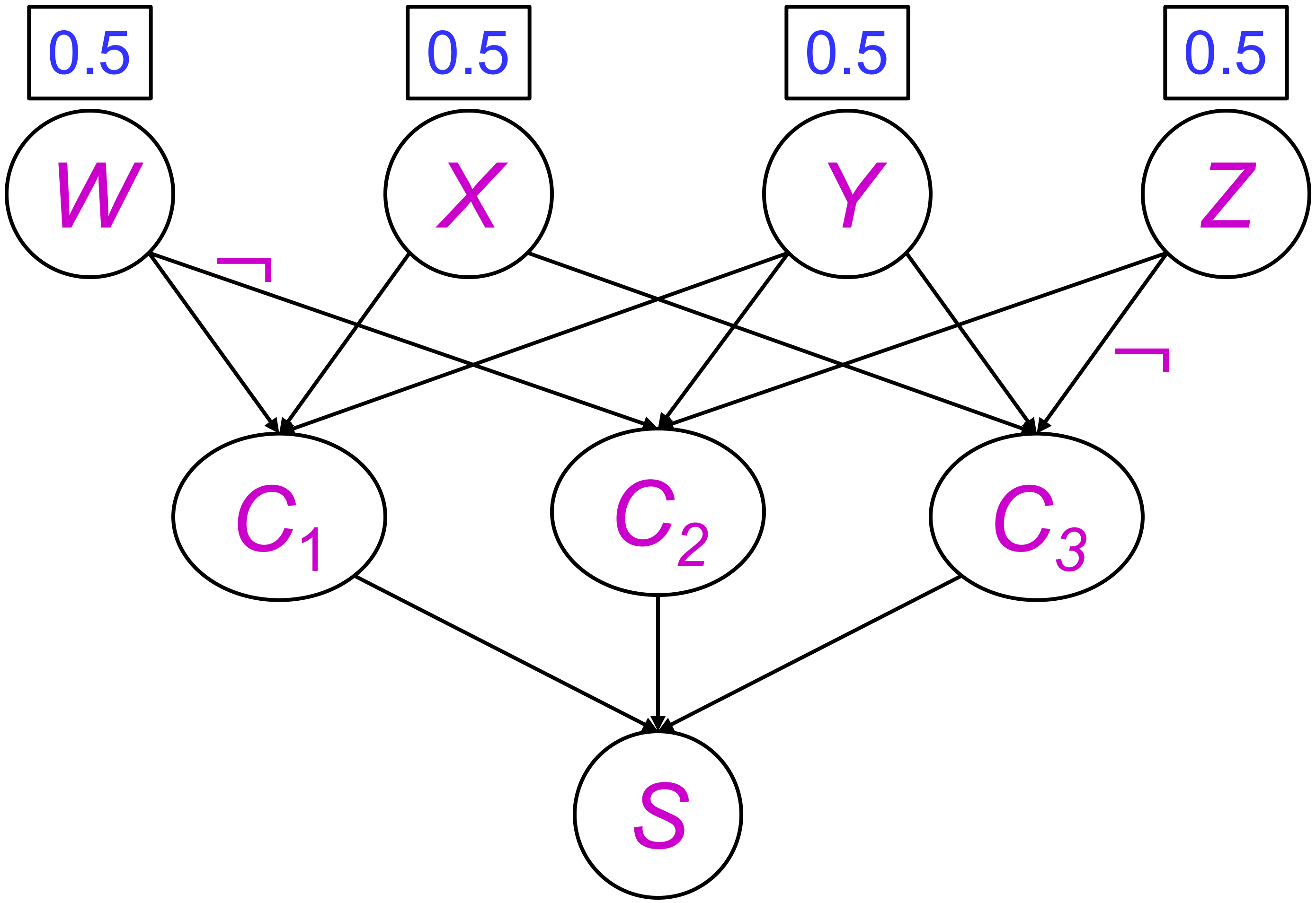
VE: Computational and Space Complexity

- The computational and space complexity of variable elimination is determined by the largest factor (and it's space that kills you)

- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example 2^n vs. 2

- Does there always exist an ordering that only results in small factors?
  - No!

Worst Case Complexity? Reduction from SAT

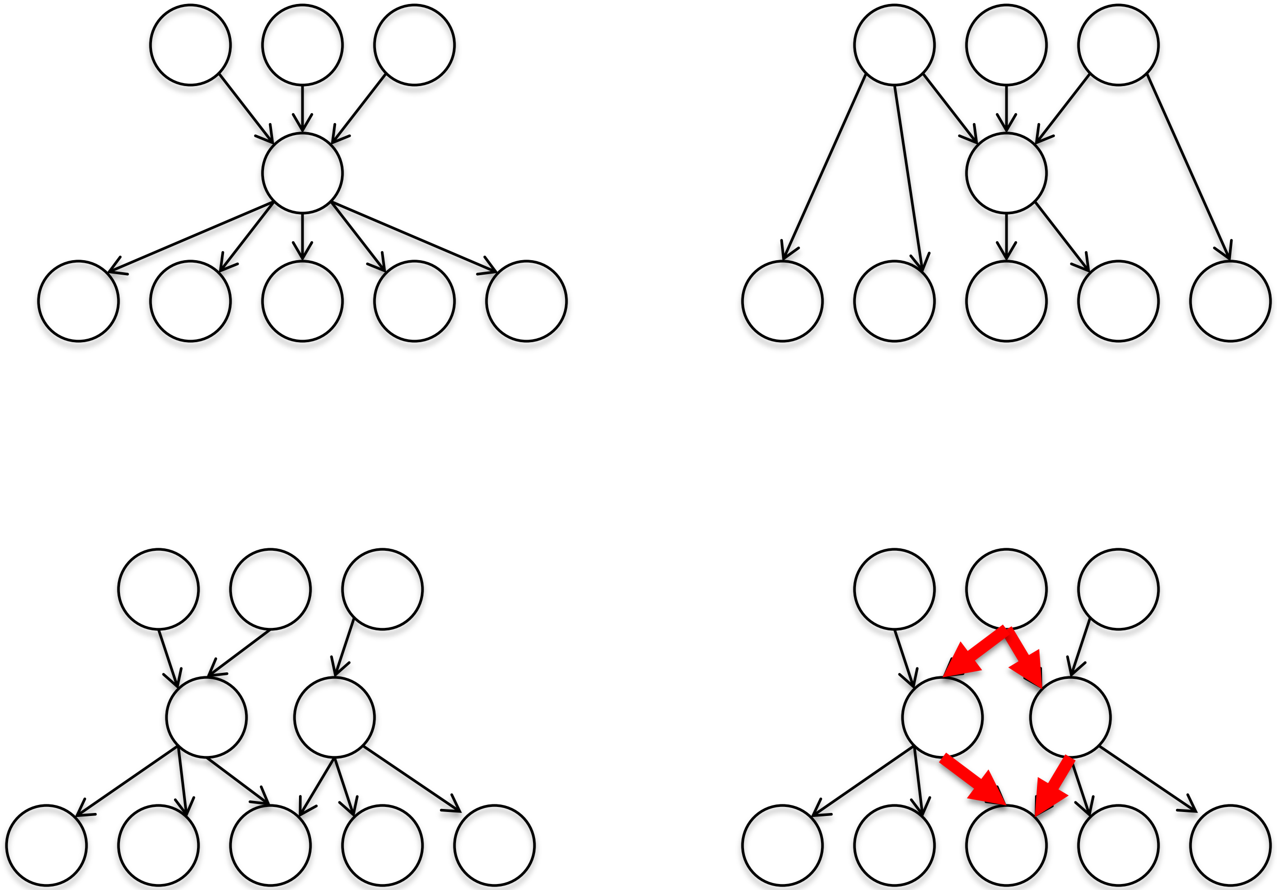


- Variables: *W, X, Y, Z*
- CNF clauses:
  1. $C_1 = W \vee X \vee Y$
  2. $C_2 = Y \vee Z \vee \neg W$
  3. $C_3 = X \vee Y \vee \neg Z$
- Sentence $S = C_1 \wedge C_2 \wedge C_3$
- $P(S) > 0$ iff S is satisfiable
  - => ***NP-hard***
- $P(S) = K \times 0.5^n$ where K is the number of satisfying assignments for clauses
  - => ***#P-hard***

Polytrees

- A polytree is a directed graph with no undirected cycles

- For poly-trees the complexity of variable elimination is ***linear in the network size*** if you eliminate from the leave towards the roots



Bayes Nets

- ✓ Part I: Representation
- ✓ Part II: Exact inference
  - ✓ Enumeration (always exponential complexity)
  - ✓ Variable elimination (worst-case exponential complexity, often better)
  - ✓ Inference is NP-hard in general
- Part III: Independence
- Part IV: Approximate Inference